Here is a list of basic Git commands along with their use cases and examples to help you use Git effectively:

1. `git init`

Use Case: Initialize a new Git repository.

Example:

```
git init
```

2. `git clone`

Use Case: Clone a repository from a remote source.

Example:

```
git clone https://github.com/example/repository.git
```

3. `git add`

Use Case: Stage changes for commit.

Example:

```
git add file.txt
```

4. `git commit`

Use Case: Commit staged changes.

Example:

```
git commit -m "Add file.txt"
```

5. `git status`

Use Case: View the status of your working directory.

Example:

```
git status
```

6. `git pull`

Use Case: Fetch changes from a remote repository and merge them into the current branch.

Example:

```
git pull origin master
```

7. `git push`

Use Case: Push local changes to a remote repository.

Example:

```
git push origin master
```

8. `git branch`

Use Case: List, create, or delete branches.

Example:

```
git branch feature_branch
```

9. `git checkout`

Use Case: Switch branches or restore working tree files.

Example:

```
git checkout feature_branch
```

10. `git merge`

Use Case: Merge changes from one branch into another.

Example:

```
git merge feature_branch
```

11. `git log`

Use Case: Display the commit history.

Example:

```
git log
```

12. `git diff`

Use Case: Show changes between commits, commit and working tree, etc.

Example:

```
git diff HEAD~1..HEAD
```

13. `git remote`

Use Case: Manage remote repositories.

Example:

```
git remote -v
```

14. `git fetch`

Use Case: Download objects and refs from a remote repository.

Example:

```
git fetch origin
```

15. `git reset`

Use Case: Reset current HEAD to the specified state.

Example:

```
git reset --hard HEAD~2
```

These are some of the fundamental Git commands that will help you get started with version control. For more advanced usage, refer to the official Git documentation (https://git-scm.com/doc).